



A direct link to your customers

– by the number one provider  
of mobile communication solutions

## LINK Mobility

# Common Platform API Description

Premium Transaction Refunds

Version 1.2; Last updated August 28, 2020

For help, see the following link <https://linkmobility.com/support/>

The most up-to-date version of this document is available at

<http://www.linkmobility.com/developers/>

## Contents

Before you begin.....	3
Scope of this document .....	3
IP addresses .....	3
Character encoding.....	3
Data types .....	4
Refund.....	4
Refund Response .....	4
Performing a refund.....	5
Base URL.....	5
HTTP methods, statuses and actions .....	5
Basic authentication .....	5
Request refund .....	5
Result codes .....	5
Examples .....	6
cURL example.....	6
Appendix 1 .....	7
Supported TLS versions.....	7
Changelog of this document.....	8

## Before you begin

Please make sure Link Mobility Support have provided you with **Username**, **Password**, **PlatformId**, **PartnerId** and a **Gateld**. To provide a Gateld support will need a result receiver URL to which refund acknowledgements can be delivered. You will also need to know the **MessageId** of the transaction you wish to refund, this is provided to you through the API for sending messages when you do that. **Username**, **Password**, **PlatformId**, **PartnerId** can be the same as you use when submitting messages.

You will also need the **MessageId** and **MSISDN** of the message you wish to refund.

Please make an opening in your firewall if necessary so that Common can connect to your system. For a list of the addresses Common will connect from, see below. Refund results are sent asynchronously.

## Scope of this document

This document will describe the Application Programming Interface (API) to perform refunds of Premium (Overcharged) SMSes. Use this if your end-user has been billed and you wish to refund the charge to your end-user.

Actually sending the MT message is described in the document "Common\_MT\_and\_DLR", available separately.

A basic familiarity with JSON and REST is assumed.

## IP addresses

When delivering a refund result to you asynchronously, the requests will be coming from the following IP addresses:

Hostname	IP address
socks1.sp247.net	195.84.162.34
socks2.sp247.net	194.71.165.71
socks3.sp247.net	195.84.162.16
socks4.sp247.net	194.71.165.98
socks5.sp247.net	195.84.162.3
socks6.sp247.net	194.71.165.122

Please add openings in your firewalls so that these six hosts can connect to you, if necessary.

## Character encoding

All communication to and from Common will be encoded using **UTF-8**.

## Data types

### Refund

Parameter	Data type	Description
<b>messageld</b>	String	The messageld of the message that should be refunded
<b>gatelds</b>	List<String>	A list of gatelds where the callbacks should be delivered when the refund is processed.

### Refund Response

The following object is returned to you when the refund is processed. Usually you are only interested in the messageld and resultCode fields.

Parameter	Data type	Description
<b>refId</b>	String	A reference id. Usually null.
<b>id</b>	String	The messageld of the message processed
<b>operator</b>	String	The telecom operator. Usually null.
<b>sentTimestamp</b>	Datetime	The timestamp when the refund was requested.
<b>timestamp</b>	Datetime	The timestamp when the refund was processed.
<b>resultCode</b>	Integer	The result of the refund request. See table below for result codes.
<b>operatorResultCode</b>	String	If the operator supplied more information, it will be here. Usually null.
<b>segments</b>	int	The number of segments that were in the transaction that was refunded. For more information, see the MT documentation.
<b>gateCustomParameters</b>	List<String>	Any custom parameters attached to the gateld supplied.
<b>customParameters</b>	List<String>	Any custom parameters used to process the request.

## Performing a refund

Base URL

**https://wsx.sp247.net/sms/refund**

### HTTP methods, statuses and actions

Method	Request ok	Invalid request Already requested	Invalid login	No access
<b>POST</b>	<b>200 OK</b> No response	<b>400 Bad Request</b> Returns Error Response	<b>401 Unauthorized</b> Returns Error Response	<b>403 Forbidden</b> Returns Error Response

### Basic authentication

For all requests; set the header “Authorization” to “Basic [encoded string]” where [encoded string] is a base 64 encoded string of your username and password separated by a colon.

### Request refund

POST /platform/[platformId]/partner/[partnerId]/msisdn/[msisdn]

URI Parameter	Data type	Description
<b>platformId</b>	String	Provided by Link Mobility.
<b>partnerId</b>	String	Provided by Link Mobility.
<b>msisdn</b>	String	The MSISDN of the transaction to refund. International format, no leading “+”.

POST the **refund** object described in the **Refund Data Type** table above.

### Result codes

Result code	Description
<b>16050</b>	Message already queued for refund
<b>2300</b>	Refund OK
<b>2301</b>	Missing or illegal MSISDN for refund
<b>2302</b>	Missing or illegal messageld for refund
<b>2303</b>	Message is queued for refund
<b>2304</b>	Refund timeout failure
<b>2305</b>	Refund failure

## Examples

Substitute your own **messageId**, **gateId**, **username**, **password**, **platformId**, **partnerId** for the ones given in the example.

### cURL example

```
curl --user "yourUsername:yourPassword" \
https://wsx.sp247.net/sms/refund/platform/yourPlatformId/partner/yourPartnerId/msisdn/4673555555 --data
'{"messageId":"EDCRkibXW/YAAAFSrLw19wDe", "gateIds": ["BV1dZyQt"]}'
```

When the refund has been processed, your callback URI (in the gateId) will receive a POST looking like this (Additional custom properties will come depending on country/segment):

```
{"refId":null,"id":"EDCRkibXW/YAAAFSrLw19wDD","operator":null,"sentTimestamp":"2016-02-04T13:03:06Z","timestamp":"2016-02-04T13:12:16Z","resultCode":2300,"operatorResultCode":null,"segments":1,"gateCustomParameters":null,"customParameters":{"gateIds":"xxxxxx"}}
```

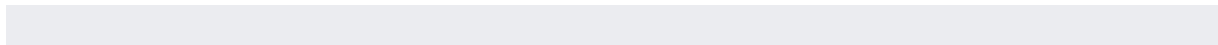
## Appendix 1

### Supported TLS versions

From 2020-11-15 will TLS 1.2 or higher be required for all HTTPS connections.

Support for TLS 1.0 and 1.1 will be discontinued. Versions 1.0 and 1.1 of TLS are older protocols that have been deprecated and are considered as security risks in the Internet community.

LINK strongly recommend to use HTTPS if HTTP is being used today. HTTP is deprecated as of 2020-09-01 by LINK, and will be removed in the future. Date for HTTP removal is not yet decided.



## Changelog of this document

<b>Date</b>	<b>Version</b>	<b>Author</b>	<b>Changes</b>
<b>2016-02-04</b>	1.0	BMS	Initial version
<b>2020-01-15</b>	1.1	EP	More refund result codes added
<b>2020-08-28</b>	1.2	JS	Added TLS Appendix, fixed footer